

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of
The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 02081258 A

(43) Date of publication of application: 22.03.90

(51) Int. Cl

G06F 15/16

G06F 15/80

(21) Application number: 63234545

(22) Date of filing: 19.09.88

(71) Applicant: FUJITSU LTD

(72) Inventor: SHINDO TATSUYA
KAWAMURA KAORU
UMEDA MASANOBU
SHIBUYA TOSHIYUKI
MITO HIDEKI

(54) PARALLEL COMPUTER

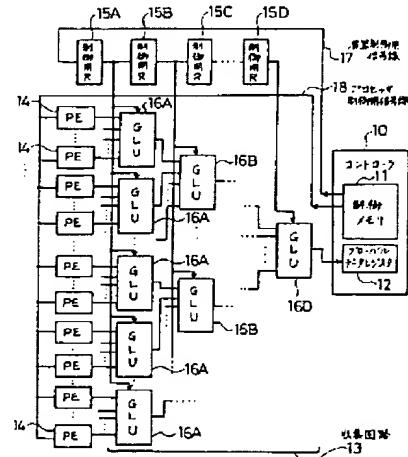
(57) Abstract:

PURPOSE: To efficiently process an operation which necessitates the general will of all processors by collecting output data sent from each processor to a controller through an arithmetic circuit by setting control information to an operation control use register.

CONSTITUTION: For instance, when a control signal of an addition instruction is set to an operation control use register 15A, a collection arithmetic circuit 16A executes addition related to output data of each processor 14, and outputs its result to a collection arithmetic circuit 16B of the next stage. By the next clock, an addition instruction of the operation control use register 15A is shifted to an operation control use register 15B, and the collection arithmetic circuit 16B executes an adding operation in the same way. Such an operation by a pipeline control advances, and in the end, when a collection arithmetic circuit 16D executes addition by an addition instruction which is set to an operation control use register 15D, its result is written in a global data register 12. In such a way, an operation which necessitates the general will of all

processors, such as taking the synchronization, etc., can be executed efficiently.

COPYRIGHT: (C)1990,JPO&Japio



⑫ 公開特許公報 (A)

平2-81258

⑬ Int. Cl. 5

G 06 F 15/16
15/80

識別記号

3 9 0

府内整理番号

Z
6745-5B
7056-5B

⑭ 公開 平成2年(1990)3月22日

審査請求 未請求 請求項の数 1 (全10頁)

⑮ 発明の名称 並列計算機

⑯ 特 願 昭63-234545

⑰ 出 願 昭63(1988)9月19日

⑮ 発明者 進藤 達也 神奈川県川崎市中原区上小田中1015番地 内 富士通株式会社

⑮ 発明者 河村 薫 神奈川県川崎市中原区上小田中1015番地 内 富士通株式会社

⑮ 発明者 梅田 政信 神奈川県川崎市中原区上小田中1015番地 内 富士通株式会社

⑮ 発明者 濵谷 利行 神奈川県川崎市中原区上小田中1015番地 内 富士通株式会社

⑯ 出願人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

⑰ 代理人 弁理士 小笠原 吉義 外2名

最終頁に続く

明細書

1. 発明の名称 並列計算機

上記各プロセッサから上記コントローラへの出力データを、上記演算制御用レジスタへの制御情報の設定により、上記演算回路を介して収集するようにしたことを特徴とする並列計算機。

2. 特許請求の範囲

処理対象となるデータを記憶する回路と演算回路とをそれぞれ有する複数のプロセッサ(14)と、これらのプロセッサを制御するコントローラ(10)とを備え、コントローラから送出する一連の命令によって、各プロセッサに与えられたデータを処理する並列計算機において、

木構造状に結合された複数の演算回路であって、最初の段では上記各プロセッサの出力を、2段目以降では前段の演算回路の出力を入力し、演算結果を後段の演算回路または最後の段では上記コントローラに出力する演算回路(16A, 16B, ...)と、

木構造の段数に応じてグループ化された上記演算回路に対し、それぞれ演算制御信号を供給する木構造の段数分のバイオペーライン接続された演算制御用レジスタ(15A, 15B, ...)とを備え、

3. 発明の詳細な説明

(概要)

SIMD型計算機等において、多数のプロセッサの総意や同期をとる収集回路を設けた並列計算機に關し、

全プロセッサが出力するデータを、必要な演算を施しつつ、効率よく収集することができるようすることを目的とし、

木構造状に結合された複数の演算回路であって、最初の段では各プロセッサの出力を、2段目以降では前段の演算回路の出力を入力し、演算結果を後段の演算回路または最後の段ではコントローラに出力する演算回路と、木構造の段数に応じてグループ化された上記演算回路に対し、それぞれ演算制御信号を供給する木構造の段数分のバイオペーラ

イン接続された演算制御用レジスタとを備え、各プロセッサからコントローラへの出力データを、演算制御用レジスタへの制御情報の設定により、演算回路を介して収集するように構成する。

〔産業上の利用分野〕

本発明は、S I M D (Single Instruction stream Multiple Data stream) 型計算機等において、多数のプロセッサの総意や同期をとる収集回路を設けた並列計算機に関する。

例えば、配線処理等のL S I - C A D を含む大規模組み合わせ問題は、非常に高速な演算能力を必要とする。これを解決するためには、多数のプロセッサを用いて構成する超並列計算機によって、高速演算処理を行うことが有望であり、多数のプロセッサを効率よく動作させる技術が必要となる。

〔従来の技術〕

ある目的のために、多数のプロセッサを動作させる場合、それらの全プロセッサの同期をとること

がしばしば必要となる。従来、この同期に関する問題を解決するために、次のような方式が用いられている。

(a) プロセッサ間通信路を用いて、同期のためのデータをやりとりする。

(b) 同期専用の信号を各プロセッサが出力し、それらのワイヤド・オア (wired-OR) 論理をとり、その結果を全プロセッサが読む。

上記(a)の方式は、特に専用の回路を必要としないが、全プロセッサと通信しなければならないために、プロセッサ間が完全結合していない限り、効率が悪い。

上記(b)の方式は、プロセッサ台数が数十～数百規模の場合には可能であるが、それ以上になる場合には、ワイヤド・オアのとれる入力数の制限や、オア論理をとるための遅延時間等の関係から、効率のよい同期処理の実現は不可能となる。

また、同期とは別に、次のような問題もある。

1つのプロセッサのデータを、他の全プロセッサに伝えるための効率のよい方法は、バスを使った

データ転送である。この場合、確かにデータ転送が始まってしまえば問題はないが、データ転送前に、どのプロセッサがバスを使うかを調停しなければならない。プロセッサ台数が、例えば数千～数万といった大規模になると、バス調停のためのオーバヘッドが無視できなくなる。

また、従来、全プロセッサが現在持っている値のうち最大値または最小値を得るとか、全プロセッサが持つ値の総和を求めるとかいうような場合に、統一的に効率よく処理するというような技術手段はなかった。

〔発明が解決しようとする課題〕

本発明は、同期をとるなど、全プロセッサの総意を必要とする演算を、超並列計算機において、効率よく実行できるようにすることを目的とする。また、全プロセッサから1つのプロセッサを選ぶというバスの調停等に相当する処理を、超並列計算機において、効率よく実行できるようにすることを目的とする。さらに、全プロセッサがそれぞ

れ持つデータ間の論理演算を、効率よく実行できるような手段を提供することを目的とする。

〔課題を解決するための手段〕

第1図は本発明の原理構成図である。

第1図において、10はコントローラ、11は出力制御信号の情報等を含むマイクロ命令が格納された制御メモリ、12は全プロセッサに対するデータ入出力用のグローバルデータレジスタ、13はプロセッサの出力データを収集する収集回路、14はプロセッサ (P E : Processing Element)、15Aないし15Dは演算制御用レジスタ、16Aないし16Dは収集演算回路 (G L U : Gathering Logic Unit)、17は演算制御用信号線、18はプロセッサ制御用信号線を表す。

プロセッサ14は、各々、処理対象となるデータを記憶する回路と演算回路とを持ち、コントローラ10からプロセッサ制御用信号線18を介して送られてくる制御信号による同一の命令の流れによって、各プロセッサ14に与えられたデータ

を処理する。

本発明に係る収集回路13は、演算制御用レジスタ15A～15Dと、収集演算回路16A～16Dとによって構成される。

収集演算回路16A～16Dは、各プロセッサ14からコントローラ10への出力データを、収集する処理を行う回路である。第1図に示すように、木構造状に結合され、1段目の各収集演算回路16Aは、何個かのプロセッサ14の出力データを入力し、その入力による演算結果を、2段目の収集演算回路16Bへ出力する。同様に、2段目から3段目、3段目から4段目というようにデータを収集していき、この例では収集演算回路16Dが最終段になっている。最終段の収集演算回路16Dは、全プロセッサ14の出力をまとめた結果を、コントローラ10のグローバルデータレジスタ12へ送り出す。

これらの回路は、木構造の段数に応じてグループ化されており、その各グループに対して、演算制御信号を供給する演算制御用レジスタ15A～

15Dが設けられる。演算制御用レジスタ15Aは、最初のグループに属する各収集演算回路16Aに対し、同じ演算制御信号を送る。次段の演算制御用レジスタ15Bは、各収集演算回路16Bに対し、演算制御信号を送る。以下、同様である。

演算制御用レジスタ15A～15Dは、木構造の段数分だけ直列にバイオペライン接続され、そのレジスタにセットされる演算制御信号は、コントローラ10から演算制御用信号線17を介して所定のクロックに応じて送出される。従って、例えば演算制御用レジスタ15Aに加算命令の制御信号がセットされると、収集演算回路16Aは、各プロセッサ14の出力データについての加算を行い、その結果を次段の収集演算回路16Bへ出力する。次のクロックでは、演算制御用レジスタ15Aの加算命令が、演算制御用レジスタ15Bへ移され、収集演算回路16Bが同様に加算演算を実行する。このようなバイオペライン制御による演算が進み、最終的に、演算制御用レジスタ15Dにセットされた加算命令によって、収集演算回路

16Dが加算を行うと、その結果は、コントローラ10のグローバルデータレジスタ12に書き込まれる。

(作用)

コントローラ10が、全プロセッサ14に対して指示した処理の同期を取る場合には、例えば、プロセッサ制御用信号線18を通して、各プロセッサ14における処理が完了したときに“1”を出力するように、プロセッサ14に指示を出す。そして、演算制御用信号線17に、アンド論理演算を指示する制御信号を送出する。

演算制御用レジスタ15Aにアンド論理演算の信号がセットされると、第1段目の収集演算回路16Aは、各プロセッサ14の出力のアンド論理演算を実行する。次のクロックで、2段目の収集演算回路16Bが同様にアンド論理演算を実行する。こうして、バイオペライン制御を進め、最終段の収集演算回路16Dが、アンド論理演算を実行したときに、その結果が“1”であれば、コント

ローラ10は、全プロセッサ14が“1”を出力したことを認知することができる。

全プロセッサ14の中から、ある条件を満たす1つのプロセッサを選択する処理は、次のように実行することができる。各プロセッサ14には、それぞれプロセッサを一意に識別するプロセッサ番号が、あらかじめ付与されているものとする。

まず、ある条件を満たすプロセッサ14は、そのプロセッサ番号を出力し、それ以外のプロセッサ14は、例えば“111…1”（または“000…0”）を出力するよう指示を出す。そして、収集回路13には、最小値（または最大値）を選択する演算制御信号を送る。

こうすることにより、収集回路13の出力として、条件を満たす1つのプロセッサ番号を得ることができる。条件を満たすプロセッサが複数あるとき、最小値による選択を行う場合には、小さいプロセッサ番号を持つものが優先的に選択されることになる。このように、バスの調停に相当する処理も、簡単に行うことができる。

加算やその他の演算についても、収集回路13により、同様にパイプライン化による高速処理が可能となる。

〔実施例〕

第2図は本発明のシステム構成例、第3図は本発明の実施例に係るプロセッサの構成例、第4図は本発明の実施例に係る収集演算回路の例、第5図は第4図に示す収集演算回路の外部信号説明図、第6図は本発明の実施例に係る収集回路の構成例、第7図は第6図に示す収集回路の動作説明図、第8図は本発明の実施例に係る収集演算回路の内部構成例、第9図は第8図に示す最大／最小／加算演算回路の例を示す。

第2図は本発明を適用したシステムの例を示している。第2図において、第1図と同符号のものは、第1図に示すものに対応し、20はプロセッサ14を格子状に結合したプロセッサ・アレイ、21はシーケンサを表す。

プロセッサ・アレイ20の全プロセッサ14お

よび収集回路13は、コントローラ10により制御される。コントローラ10からは、プロセッサ14内部のファシリティ単位で制御が可能であり、プロセッサ間通信の通信方向、内部におけるセレクタの入力選択、演算回路のオペレーション、データレジスタのアドレス等が制御の対象となる。

これらのファシリティへの制御用信号線により、プロセッサ14とコントローラ10とが接続され、コントローラ10の内部にある制御メモリ11に、所定の制御フォーマットで書かれた値が、各マシンサイクルごとに制御用信号線に出力される。収集回路13に対しても同様に、収集時に必要となる演算制御信号がコントローラ10から送出される。

コントローラ10におけるシーケンサ21は、制御メモリ11から制御情報を読み出すシーケンスを決定する回路である。グローバルデータレジスタ12は、全プロセッサ14に共通に送り出すデータを保持したり、収集回路13の出力結果を受け取るためのレジスタである。

第2図に示す各プロセッサ14は、第3図に示すような構成になっている。内部に、処理対象となるデータを保持するデータレジスタ30と、入出力データおよびデータレジスタ30に記憶されているデータについての演算を行う演算回路(ALU)31を持つ。

各プロセッサ14は、コントローラから送られる同一の制御信号で制御される。この制御信号には、データレジスタ30のアドレス、演算回路31に対するオペレーション・コード等が含まれる。

プロセッサ14は、東(E)、西(W)、南(S)、北(N)の4方向の隣接プロセッサとの通信を行うためのEWN Sポートを持つ。また、全プロセッサ14に対して、コントローラから同じ値を与えるためのグローバルデータ入力と、第2図に示す収集回路13への収集データ出力の端子を持つ。

第3図に示すプロセッサ14は、1ビット・プロセッサであり、データレジスタ30への入出力や外部とのデータ入出力は、すべて1ビットを基

本としている。1ビットより大きいデータは、その大きさに応じて1ビットずつ、上位ビットまたは下位ビット側から連続して処理する。これにより、何ビットのデータでも処理することができるようになっている。収集データ出力についても、1ビットである。もちろん、本発明は、この例のような1ビット・プロセッサに限らず、複数ビットの入出力を行うプロセッサにも同様に適用することが可能である。

プロセッサ14の内部構成については、種々のものが知られており、本発明の要旨とするところから外れるので、この程度の説明にとどめる。

収集演算回路16は、例えば第4図に示すような外部信号の入出力をを行う。その外部信号の詳細な内容を、第5図に示す。

本実施例の収集演算回路16は、内部に論理演算回路40と、論理積演算回路41と、最大／最小／加算演算回路42とを持つ。

GL1は、収集演算回路16への32ビットの入力データである。すなわち、1つの収集演算回

路16で、最大32個のプロセッサ14または他の収集演算回路16の収集データ出力を処理することができるようになっている。GLOは、演算結果の出力データであり、1ビットである。

制御系の信号として、以下のものがある。

(a) XGOPS (2ビット)

収集演算回路16のオペレーションコードである。これが“00”的ときアンド(AND)，“01”的ときオア(OR)，“10”的とき最小(MIN)または最大(MAX)，“11”的とき加算(ADD)の演算を行う。

(b) XGCR (1ビット)

加算時などにおいて、キャリー(内部で記憶する桁上がり等のビット)をクリアするか、クリアしないかを指示する。

(c) GLSTS (2ビット)

収集演算回路16への入力段数を切り換える。すなわち、32ビットの全部を入力するか、16ビットだけ入力するか、8ビットだけ入力するか、4ビットだけ入力するかを選択する。入力段数は、

収集演算回路16の接続構成によって静的に決まる。この例では、汎用的に収集演算回路16を用いることができるようにするために、入力段数を切り換え可能としているが、この切り換え機能はなくてもよい。

(d) GMAXS (1ビット)

XGOPSが“10”的とき、最大または最小のどちらの演算を行うかを指定する。

(e) GNOPS (1ビット)

強制0入力信号であり、このビットが“0”的とき、GLIの入力データを無効化する。

第6図は、第1段目に4個の収集演算回路16-1～16-4と、第2段目に1個の収集演算回路16-5を用いて、128入力の収集演算を行う収集回路の例を示している。

説明を簡単にするために、演算制御用レジスタ15-1、15-2におけるオペレーションコードXGOPSとキャリー制御信号XGCR以外の信号は、省略する。50～58は、それぞれバイオペライン制御のためのレジスタである。

OP1は、演算制御用レジスタ15-1にセットされた信号、OP2は演算制御用レジスタ15-2にセットされた信号、D1はレジスタ50～53にセットされたデータ、D2はレジスタ54～57にセットされたデータ、D3はレジスタ58にセットされたデータを表す。

第7図は、第6図に示す収集回路の動作を説明するためのタイムチャートである。第7図では、代表的な演算の1つとして、4ビット長のデータの加算を行うときの例を示している。

B0～B3は、処理対象となる4ビットのデータであり、B0は最下位ビット(LSB)、B3は最上位ビット(MSB)である。加算の場合、これらのデータは、下位ビット側のB0から順次B3へ1ビットずつ、各クロックごとにレジスタ50～53に入力される。

このデータに同期して、演算制御用レジスタ15-1(OP1)に、加算のオペレーションコードがセットされる。XGCRは、最初のクロックでは“0”でキャリー・クリア、以後のクロック

では“1”で通常となるように入力される。

第1段目の収集演算回路16-1～16-4は、最初にB0について、OP1により、それぞれ32ビットの加算演算を実行する。次のクロックでは、B1について、B0の加算結果によるキャリーを考慮した加算演算を実行する。以下、同様にB3まで、加算演算を繰り返す。

収集演算回路16-1～16-4の加算結果は、レジスタ54～57にセットされる。また、OP1は、毎クロック、OP2へ移される。

収集演算回路16-5は、OP2に応じて、4入力の加算演算を実行し、結果をレジスタ58に出力する。すなわち、収集演算回路16-5は、1クロック遅れて、収集演算回路16-1～16-4と同様に動作する。

以上のようにバイオペライン制御による動作が行われるようになっている。4ビット長ではなく、他のビット長の加算も、同様に連続的に実行することができる。また、最大/最小等の演算についても、任意のビット長について同様に実行可能で

ある。ただし、最大／最小の演算の場合には、ビット列について、最下位ビット側からではなく、最上位ビット側から1ビットずつ入力する。

第6図に示す回路を、さらに多段化することなどにより、128入力に限らず、多数の入力データを処理することができる。

本実施例における収集演算回路16の内部構成は、第8図に示すようになっている。

第8図において、A1はアンド回路、N1～N5はノア回路、NA1～NA5は NAND回路、S1～S5はセレクタ、MA00～MA40は最大／最小／加算演算回路を構成する回路ブロックを表す。

アンド回路A1は、強制0入力信号GNOPSが“0”的とき、入力データGL1をオール・ゼロにする。ノア回路N1～N4、NAND回路NA1は、入力データGL1について、入力段数に応じた論理和演算を行う。

NAND回路NA2～NA5、ノア回路N5は、入力データGL1について、論理積演算を行う回

路である。

MA00ないしMA40は、最大／最小／加算演算を行うための回路であり、2ビットずつ、トーナメント式に、最大／最小／加算の演算を実行するようになっている。その詳細な内部構成については、第9図に従って後述する。

セレクタS1～S4は、オペレーションコードXGOPSに応じた出力の選択を行い、セレクタS5は、入力段数に応じた出力の選択を行う。

最大／最小／加算演算回路を構成するMA00～MA40の1ブロックは、第9図に示すような回路構成になっている。

第9図において、A10～A16はアンド回路、N10～N12はノア回路、NT0～NT6はノット回路、O1～O2はオア回路、R0～R1はキャリーを記憶するレジスタ、S10～S11はセレクタ、90は3入力の加算器である。

前述のように、第9図に示す回路は、2入力の最大／最小または加算演算を実行する。DM0、DM1は、最大／最小を求めるための各1ビット

の入力信号である。DA0、DA1は加算対象となる各1ビットの入力信号である。XMは最大／最小の演算結果、XAは加算の演算結果である。最初の段、すなわち、第8図に示すMA00～MA015では、DM0とDA0、DM1とDA1は、それぞれ同じ値である。

最初に、第9図に示す回路の加算の動作について説明する。

加算のデータは、最下位ビットから1ビットずつ、DA0、DA1に入力される。最初のビットでは、XGCRが“0”であるため、キャリー・クリアとなって、アンド回路A10の出力であるキャリーCARRY0が“0”となる。DA0、DA1について、加算器90による加算が行われ、加算結果XAが出力される。加算による桁上がりがない場合、キャリー出力CARRYOUTは“0”となり、桁上がりがある場合、キャリー出力CARRYOUTは、“1”となる。

このキャリー出力CARRYOUTは、次のクロックにおける加算のために、セレクタS10を経て、レ

ジスタR0に保持される。

最下位ビットの次からは、レジスタR0の内容がキャリーCARRY0として取り出され、DA0、DA1と共に加算される。

最大値を求めるときの動作は、以下のとおりである。最大値を求める場合、信号GMAXSは、“0”である。

入力データは、最上位ビットから1ビットずつ、DM0、DM1に入力される。最初のビットでは、XGCRが“0”とされ、アンド回路A10、A11の出力が“0”となって、ノア回路N12の出力が“1”，アンド回路A14、A15、オア回路O1の出力が“0”になる。従って、セレクタS11の選択信号は、“10”となり、最大値の出力XMとして、オア回路O2の出力によるDM0とDM1の論理和が選択出力される。

簡単な論理演算によって明らかのように、レジスタR0、R1には、DM0とDM1のいずれか一方だけが先に“1”になった場合、その対応する側に“1”がセットされる。すなわち、DM0

が“1”で、DM1が“0”的場合、レジスタR0に“1”がセットされ、これとは反対に、DM0が“0”で、DM1が“1”的場合、レジスタR1に“1”がセットされる。以後、その状態を保持する。

レジスタR0、R1のいずれか一方に、“1”がくると、次のクロックからは、ノア回路N12の出力が“0”になり、また、オア回路O1の出力は、R0が“1”的場合に“1”，R1が“1”的場合に“0”になる。従って、セレクタS11は、DM0またはDM1のうち、先に“1”が検出された側の値を、それ以後、選択して出力を続ける。

最小値を求める場合、信号GMAXSは“1”である。最大値を求める場合と動作は同様であるが、ノア回路N12の出力が“1”であるときの、セレクタS11に対する選択信号は“11”であり、アンド回路A16の出力が選択される。

DM0、DM1のいずれか一方が“1”になると、それに対応してR0、R1の一方が“1”に

セットされ、セレクタS11に対する選択信号は、“00”または“01”になる。以後、DM0またはDM1の最小値側が選択されて出力される。

第9図に示す回路を、第8図に示すように5段重ねれば、32ビットの入力に対する最大／最小または加算の演算を実行できることになる。入力段数が、例えば8になっている場合には、第8図に示す3段目の出力XM8またはXA8を取り出せば、求める演算結果を得ることができる。

第2図に示す並列計算機は、例えばLSI設計における配線処理等に用いることができる。配線処理を行うための代表的な方法として、迷路法があるが、収集回路13を用いることにより、このような迷路法の処理において必要となる以下の機能の実現が可能になる。

- ① 各処理の終了判定、すなわち、プロセッサ全体の同期制御。
- ② 複数ターゲットにおけるコストの最小値計算。
- ③ クロス・オーバラップしたネット番号の読み出しなど、ある条件を満たしているもののプロ

セッサ番号等の読み出し。

- ④ クロス・オーバラップした全ネット数の加算など、ある条件を満たしたもののかウント処理。もちろん、本発明に係る並列計算機を、他にも種々の用途に適用して実施することが可能である。

(発明の効果)

以上説明したように、本発明によれば、全プロセッサが出力するデータを、必要な演算を施しつつ、効率よく収集することができるようになり、 SIMD型の多数のプロセッサを持つ並列計算機等を有効に利用できるようになる。また、機能拡張により、並列計算機の適用対象となる分野の拡大が可能になる。

4. 図面の簡単な説明

第1図は本発明の原理構成図。

第2図は本発明のシステム構成図。

第3図は本発明の実施例に係るプロセッサの構成図。

第4図は本発明の実施例に係る収集演算回路の例。

第5図は第4図に示す収集演算回路の外部信号説明図。

第6図は本発明の実施例に係る収集回路の構成例。

第7図は第6図に示す収集回路の動作説明図。

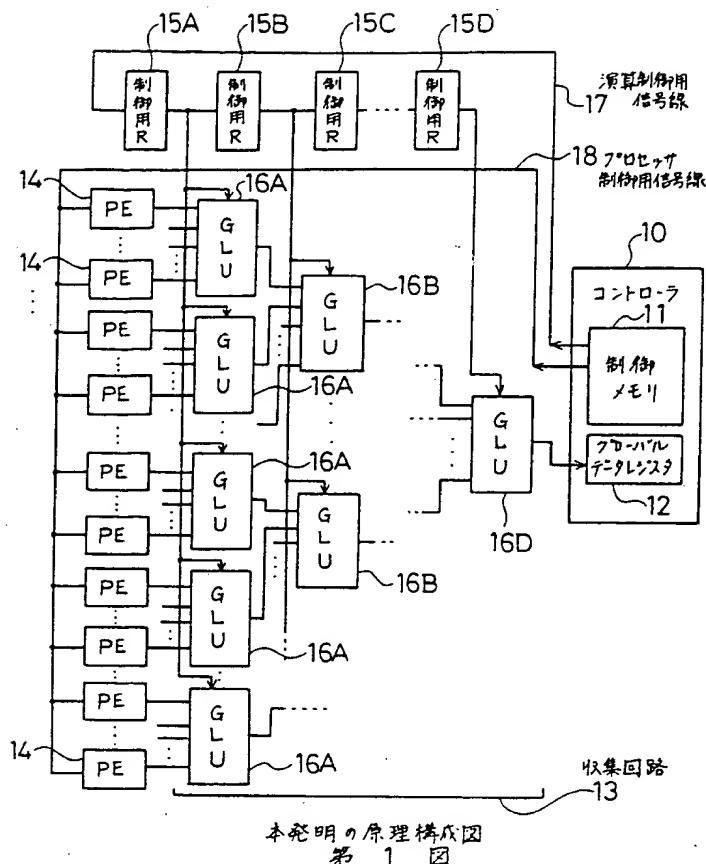
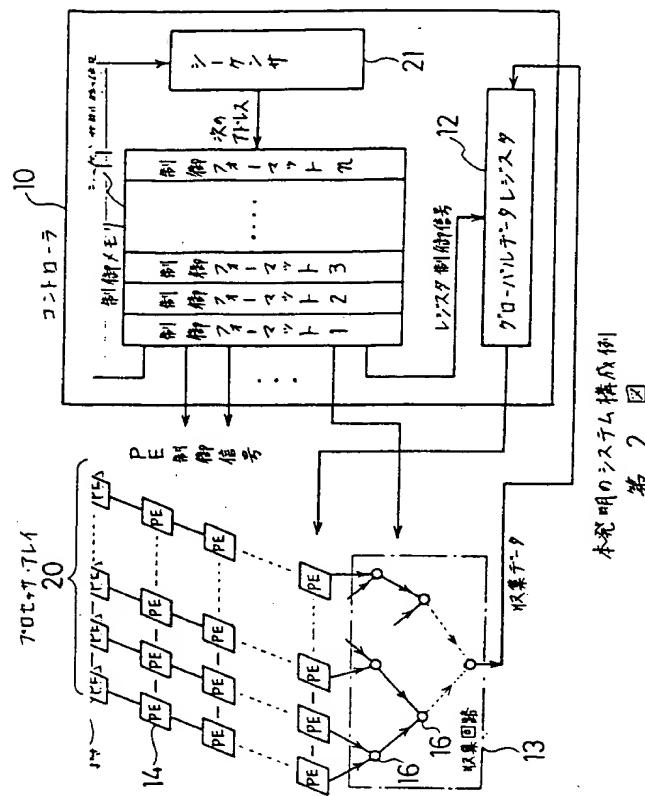
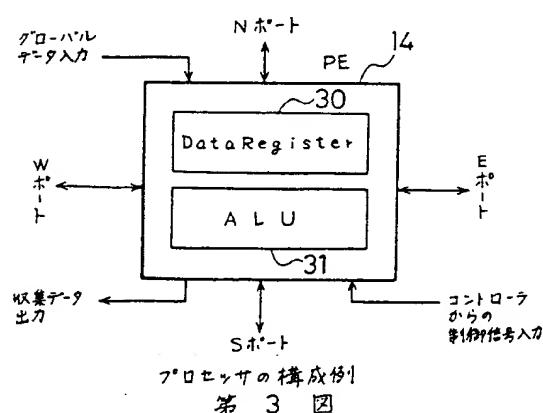
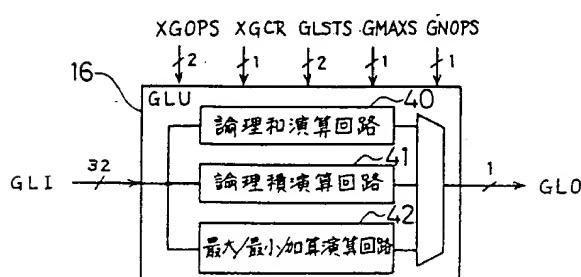
第8図は本発明の実施例に係る収集演算回路の内部構成例。

第9図は第8図に示す最大／最小／加算演算回路の例を示す。

図中、10はコントローラ、11は制御メモリ、12はグローバルデータレジスタ、13は収集回路、14はプロセッサ、15A～15Dは演算制御用レジスタ、16A～16Dは収集演算回路、17は演算制御用信号線、18はプロセッサ制御用信号線を表す。

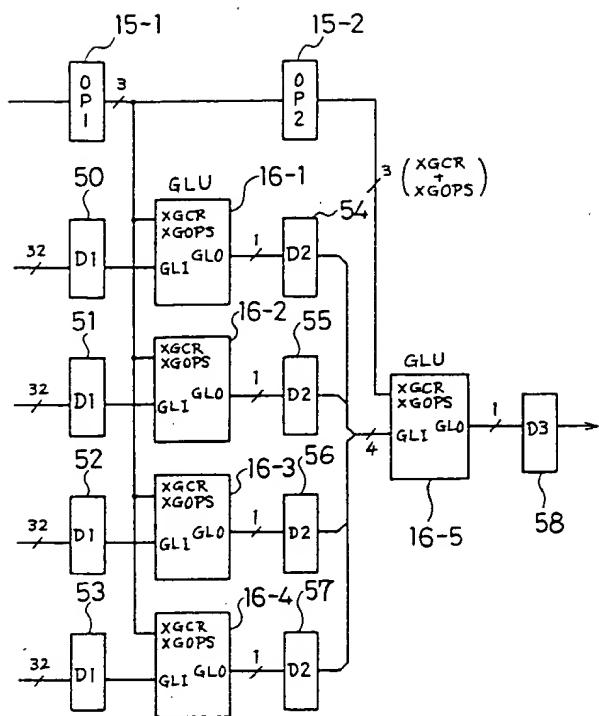
特許出願人 富士通株式会社

代理人 弁理士 小笠原吉義（外2名）

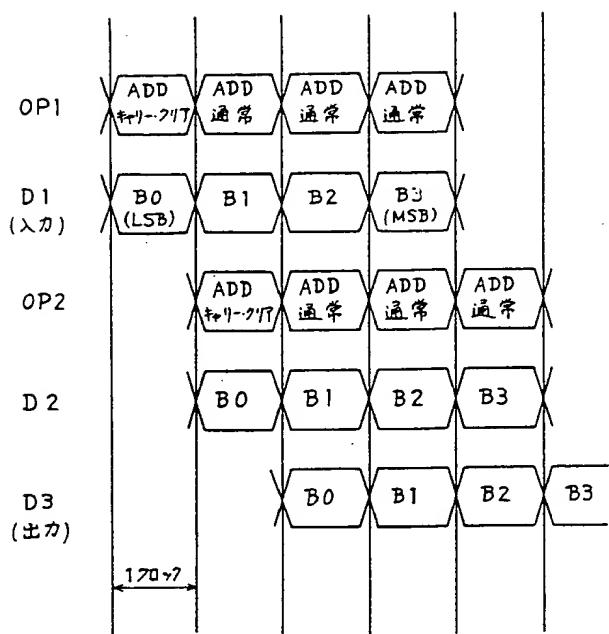
本発明の原理構成図
第1図本説明のシステム構成例
第2図プロセッサの構成例
第3図收集演算回路の例
第4図

信号名	bit	属性	意味
GLI	32	1	GLU 入力データ
GLU	1	0	GLU 出力データ
計	33		
XGOPS	2	1	GLU オペレーショントラップ (00:AND, 01:OR, 10:MIN/MAX, 11:ADD)
XGCR	1	1	キャラリーアップ (0:キャラリーアップ, 1:通常)
GLSTS	2	1	GLU入力段数印板 (00:32, 01:16, 10:8, 11:4)
GMAXS	1	1	MAX,MIN選択信号 (0:MAX, 1:MIN)
GNOPS	1	1	強制入力信号 (0:0入力, 1:通常)
計	7		
計 40 bit			

收集演算回路外部信号説明図
第5図



収集回路の構成例 第 6 図



収集回路の動作説明図

第 7 圖

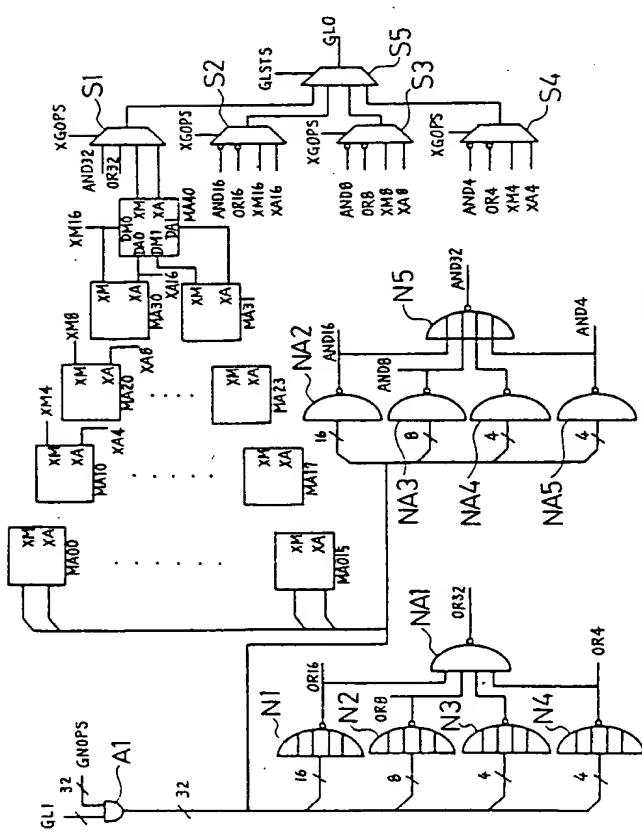
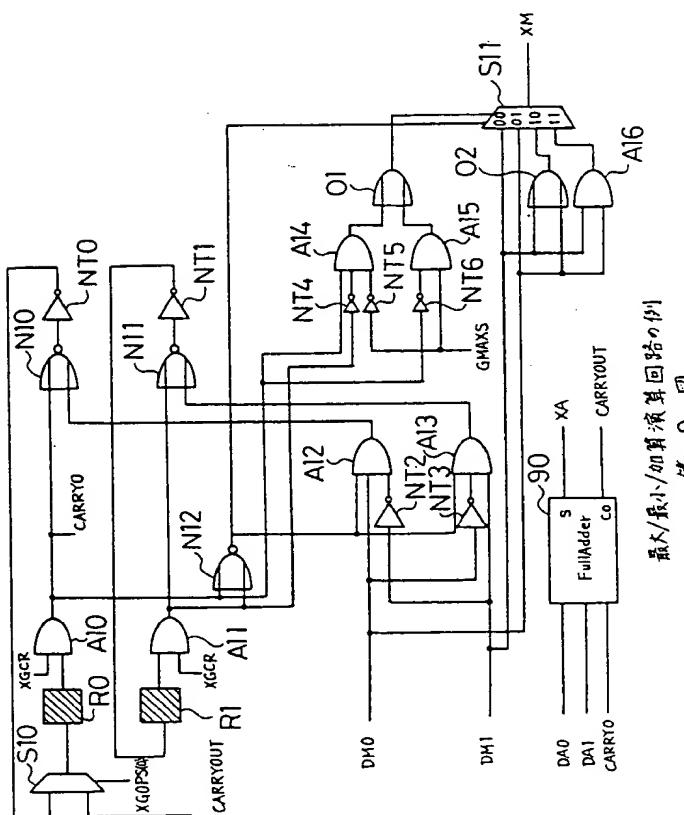


圖 8 第 8 演算回路の内部構成例



最小/加算演算回路の例

第1頁の続き

⑦発明者 三渡 秀樹 神奈川県川崎市中原区上小田中1015番地 富士通株式会社
内